

ネットワークオペレーションにおける 生成AI技術の活用検討について

NTTフィールドテクノ
佐藤 亮介 白井 嵩士 田口 順史 近藤 優吉



- ・ 名前: 佐藤 亮介
- ・ 現担当: NW運用/保守
- ・ JANOG歴: 登壇2回目
- ・ 趣味: ITガジェット, 仮想通貨



- ・ 名前: 白井 嵩士
- ・ 現担当: NW運用/保守
- ・ JANOG歴: 初登壇
- ・ 趣味: 局舎巡り, 川魚のお世話



- ・ 名前: 田口 順史
- ・ 現担当: NW運用/保守
- ・ JANOG歴: 初登壇
- ・ 趣味: トロンボーン, 枯山水

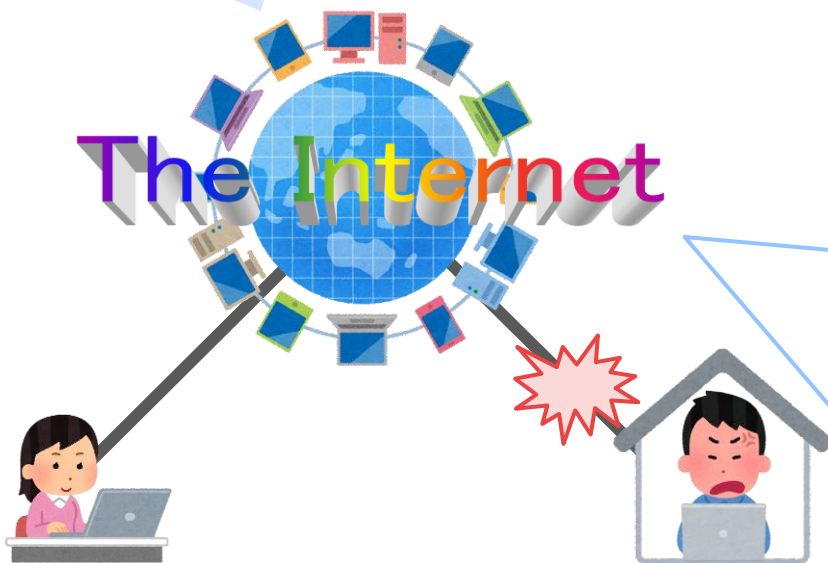


- ・ 名前: 近藤 優吉
- ・ 現担当: NW運用/保守
- ・ JANOG歴: 初登壇
- ・ 趣味: バイク

- リモートワークの普及や大規模なネットワーク障害を通してNW保守の重要性が高まっておりNTTフィールドテクノのNOCでは日々迅速な障害復旧に向けて努めています。

- ・トラフィックの増加
- ・インターネットへの期待

UP

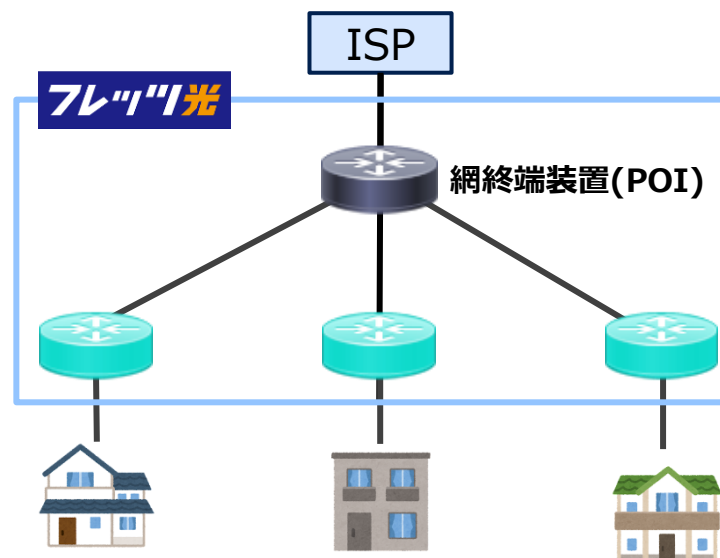


NTTフィールドテクノのNOCでは・・・

① NTT西日本サービスにおける
NWの運用保守を実施！

② 迅速な障害復旧に向けた
オペレーション高度化を推進！

運用装置数は数十万台規模！

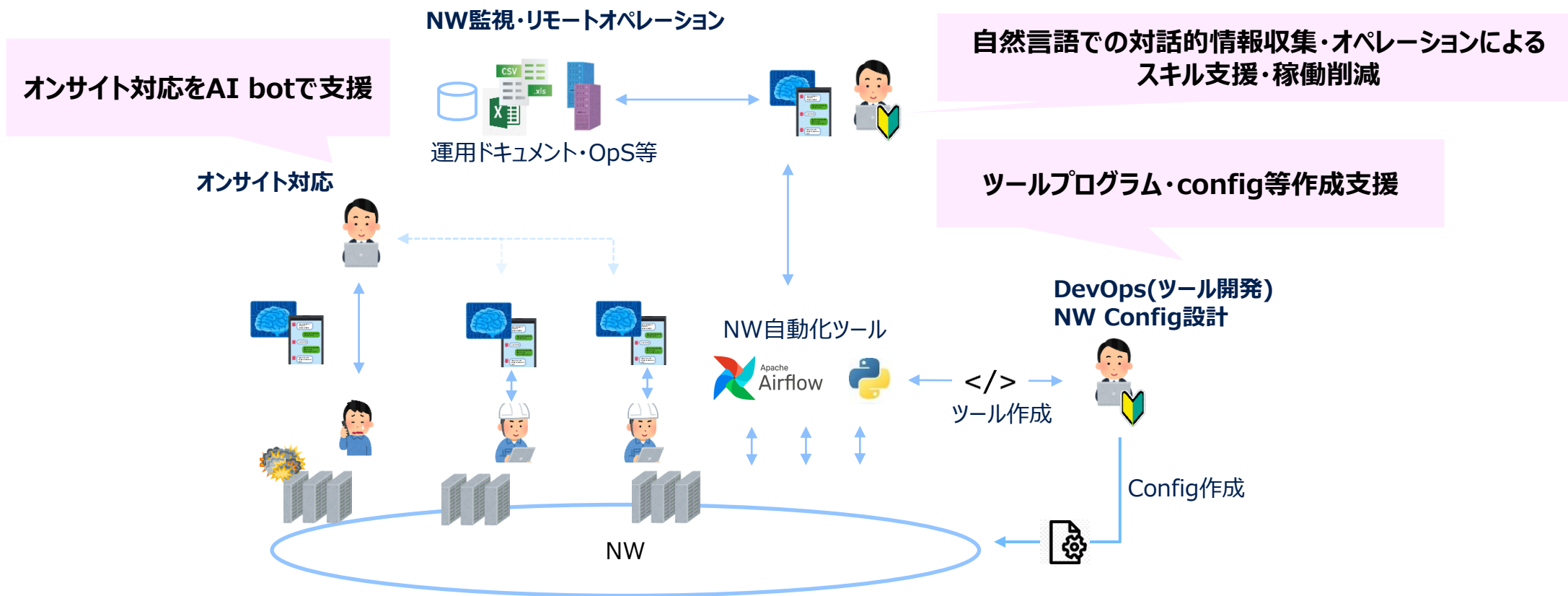


保守に加えて
NW自動化等のDevOpsも実施



- 生成AIが急速に普及しているが、**ネットワークオペレーション特有の業務への活用には至っていない状況**。
- 各種オペレーション業務の生産性向上が見込まれるとともに将来的なオペレーション業務デザインに大きく影響する要素のため本格的な検討に着手している。

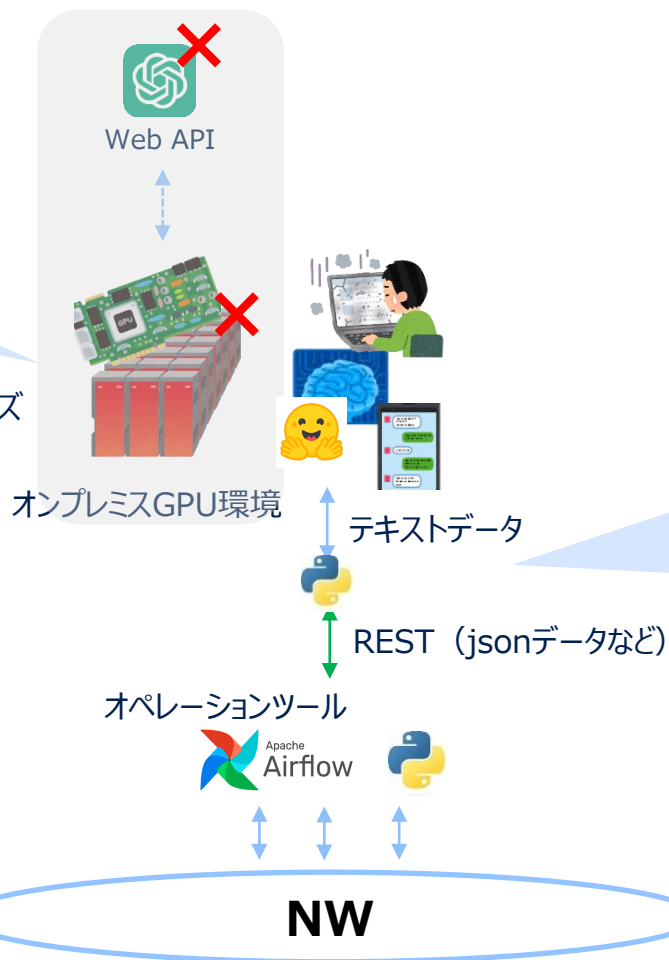
ネットワークオペレーション業務への生成AI活用イメージ(例)



- 取り組みの中で抽出された課題を示す。
- これらの課題はすべてのユースケースに共通する基礎的なものであるため、解決のアプローチを紹介し議論させていただきたい。

課題①: LLMの動作環境構築

クラウド上の大規模言語モデル（LLM）の利用や、オンプレミスによるGPU環境構築がセキュリティ・コスト面でハードルが高い



課題②: 専門情報・ローカル情報との連携

オペレーション判断に必要な情報の多くが専門的かつ外だしできず、頻繁に更新されるため LLMに事前学習させられない

課題③: オペレーションツールとの連携

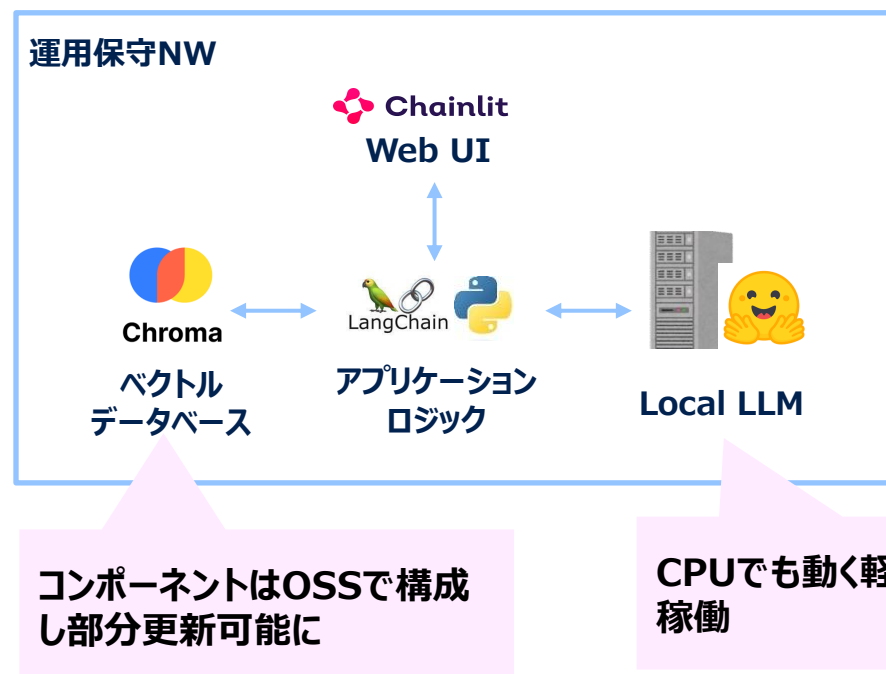
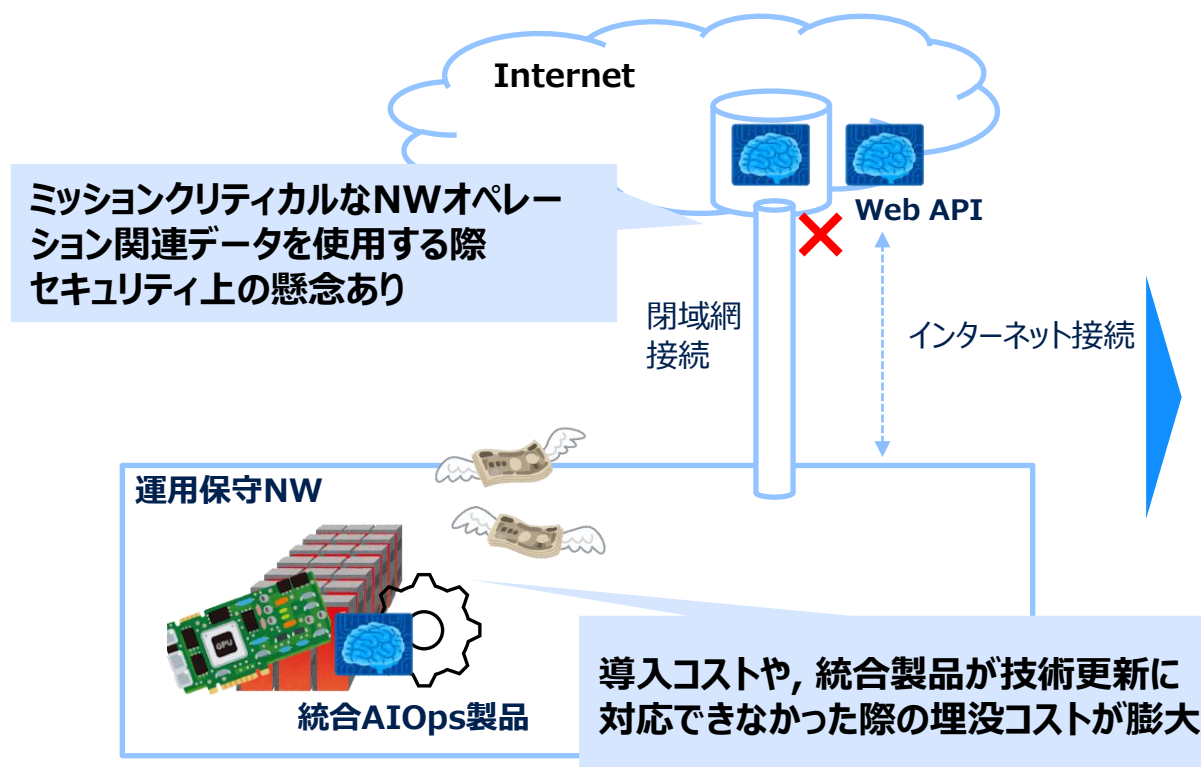
NW装置へのオペレーションを実行するには、テキストから具体的なツールの実行を判断し入力データを生成する（json等）実行結果が文意を満たしているか判断するなど連携モジュールの作りこみが必要

🤗 : 生成AIのコミュニティ（Huggingface Hub）のアイコンで生成AIモデルを示すのに使っています

課題①：LLMの動作環境構築に関する課題とアプローチ

6

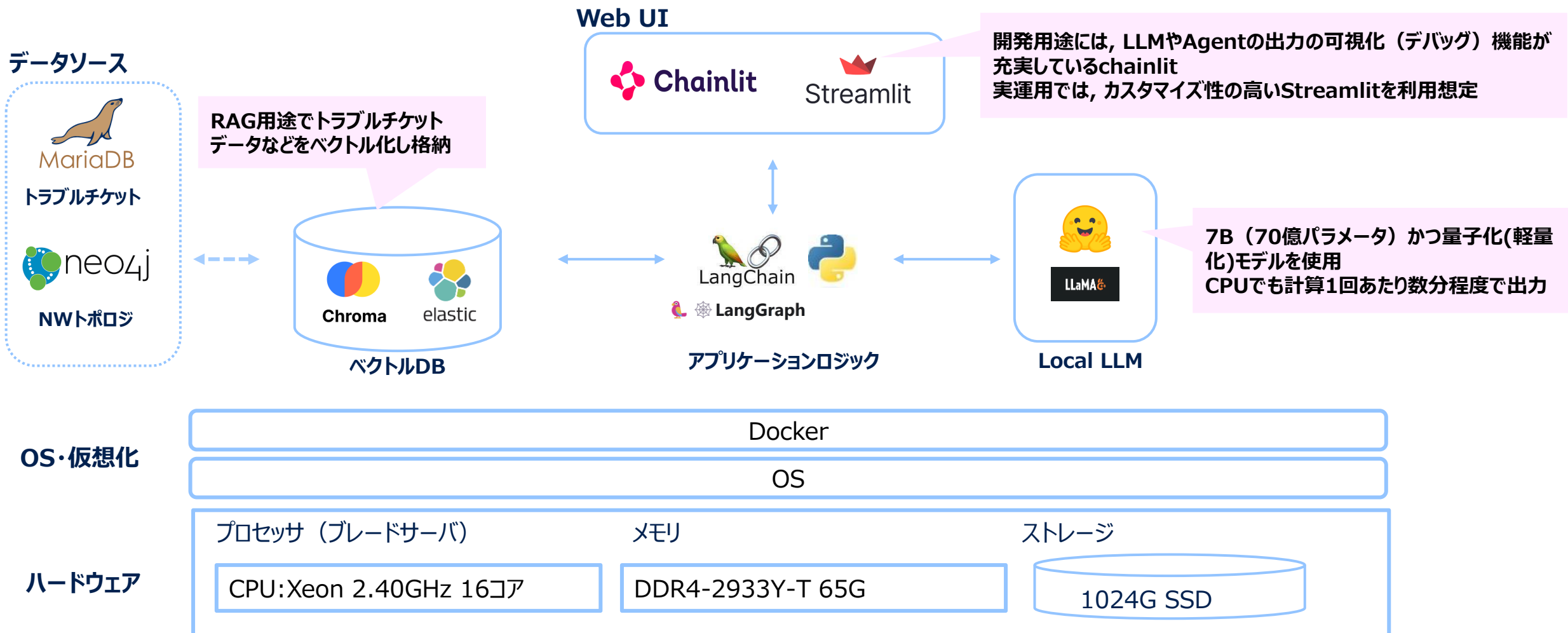
- LLMの環境構築に際し, chatGPTなどのWeb APIは**セキュリティ上のハードル高いケースがある**.
- 市中の統合AIOps製品をオンプレ環境に導入する場合も, GPU等要求マシンスペックが高く**導入コストがかさみ**, コンポーネント同士が密結合になりやすくAIの技術方式の更新の際に**埋没コストが膨大**になるリスクが大きい.
- そこで, セキュアなオンプレ環境で動作する軽量な**Local LLM**と**OSS・ライブラリ**を疎結合に組み合わせたアーキテクチャにて様々な技術方式を試行錯誤しながら生成AIの活用ノウハウを蓄積している.



アーキテクチャ詳細

7

- 各コンポーネントごとに、複数の実装方式（適用OSS・モデル）を採用し拡張性を担保している。

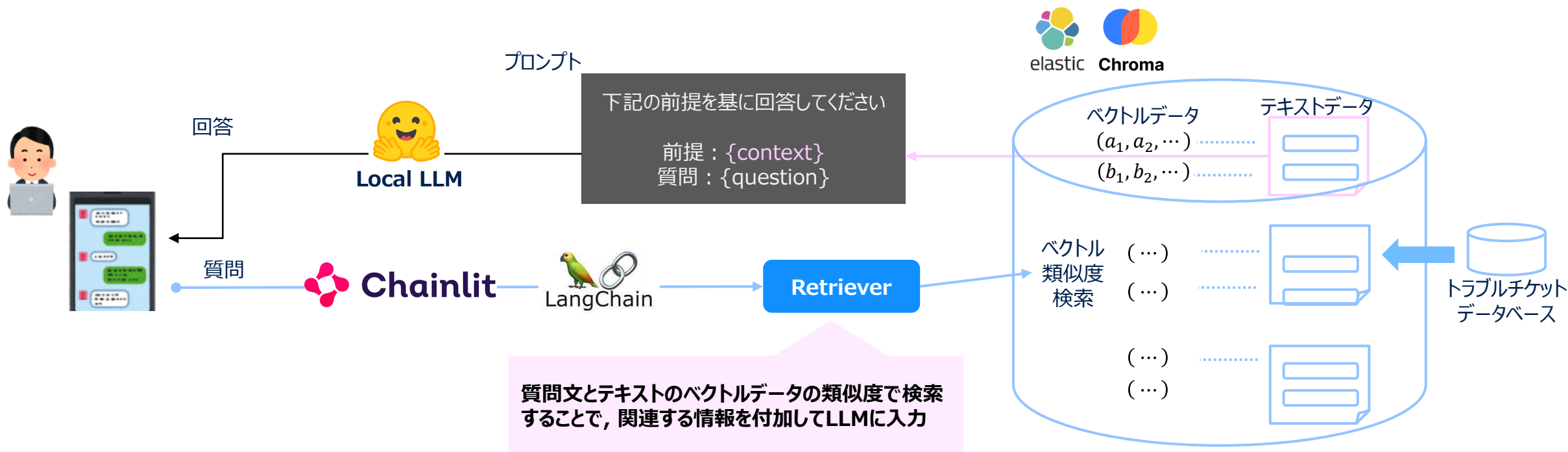


課題②：専門情報・ローカル情報との連携に関する課題とアプローチ

8

- NWオペレーションにおける判断には、過去のトラブルチケットの内容など専門的な情報が必要になるが、頻繁に更新されるこれらの情報を**Local LLMに直接学習(ファインチューニング)**させることは難しい。
- ベクトルデータベース化した情報を検索しLLMの入力に付加する**RAG(Retrieval-Argumented Generation)**という手法を用いてNWオペレーション特有の専門的情報をLocal LLMが取り扱えるようにした。

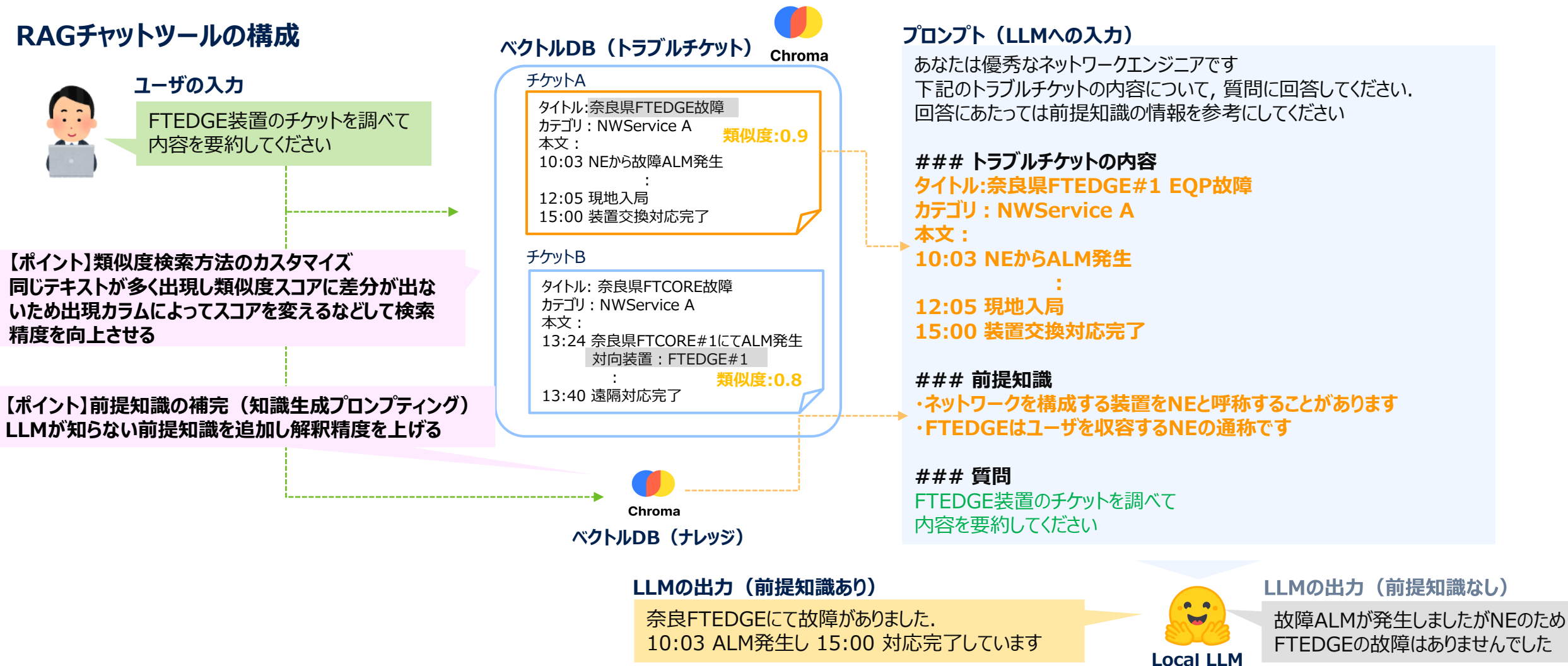
RAGアーキテクチャイメージ



トラブルチケットのRAGチャットツールの構成例と技術ポイントの紹介

- トラブルチケット(故障対応の履歴)の情報をRAGを用いて扱えるようにしたチャットツールの構成例を示す.
- 単純にベクトルDBからトラチケのテキストを検索するだけでは, 検索精度及びLLMの解釈精度に難があるため **類似度検索方法のカスタマイズ・前提知識の補完**などのチューニングを行うことが必要.

RAGチャットツールの構成



RAGツールの出力傾向

- チケット単体に対する情報抽出・要約に関する単純な質問については妥当な回答を行えることが確認できた。
- 複数チケットに跨る質問や生成・推論のタスクを含む質問については、現状のアーキテクチャでは正しい回答が生成できない。

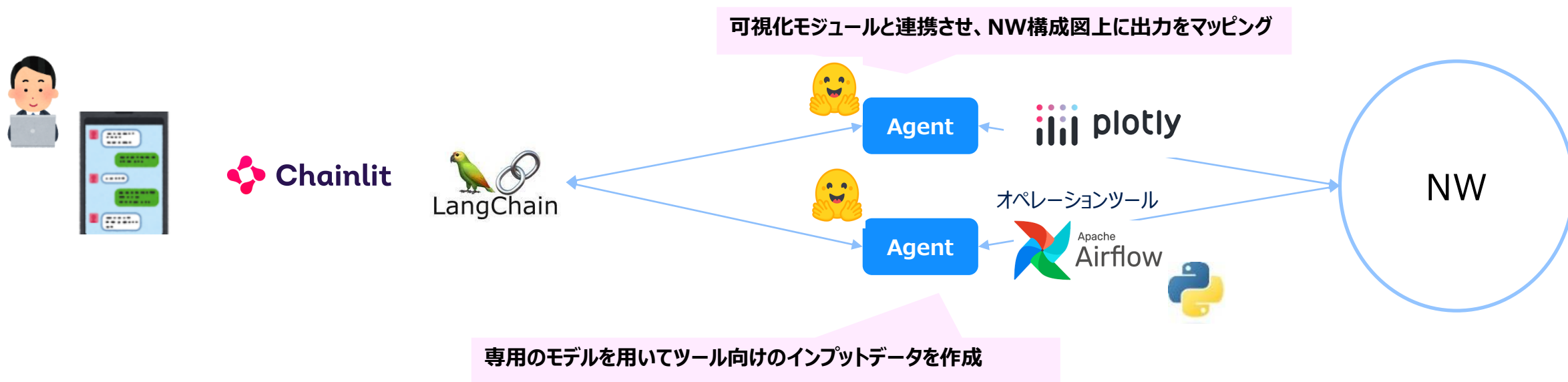
分類	プロンプト例	出力傾向
チケット単体の情報抽出・要約	・FTEDGE装置のチケットを調べて、故障したポートを教えてください ・FTEDGE装置のチケットを調べて内容を要約してください	・装置名など、プロンプト中に具体的なキーワードを組み込んだ場合は妥当なチケットがRetrieveできることが多い ・正しい装置部位や誤りでない要約を返答する
複数チケットに跨る情報抽出・要約	・FTEDGE装置の故障最頻エリアを教えてください ・ここ半年間のユーザ影響のあった故障数を教えてください	・ロジック上複数チケットをRetrieveできていないため、誤り・不自然な回答になる
生成・推論	・FTEDGE装置のRPが故障した場合の対処手順と注意事項を教えてください ・作業履歴からオペレータごとの故障対応スキルを評価し、課題を見つけてください	・一般常識やプロンプトに与えた前提知識からでは推論できないため、誤り・不自然な回答になる

課題③：オペレーションツールとの連携に関する課題とアプローチ

11

- LLMのテキスト形式のアウトプットを実際のオペレーションにつなげるためには、**オペレーションツールに必要なインプットデータを作成**するなど、LLMと既存のツールを連携させるモジュールの作りこみが必要。
- langchain Agentを利用し、ワークフローエンジン（装置へのCLIオペレーション）及びNW可視化ツールとの連携をモジュールを作成した。

Agentアーキテクチャイメージ

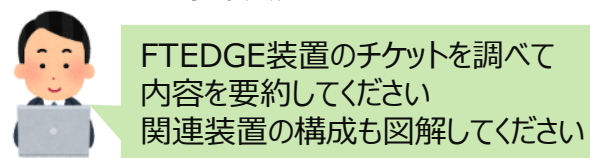


オペレーションツールとLLM連携事例紹介①：NW可視化ツールとの連携

- CLIオペレーションを実行するワークフローエンジンの実行に必要な入力パラメータをLLMに生成させ、応答にNW構成の図解を付記することを可能とした。
- インプットデータの生成のプロンプトには**出力形式の指定**を行うことが、動作の正確性を向上させるために必要となる。

可視化ツール連携Agentの構成例

ユーザの入力



トラブルチケットRAG
Agent



【ポイント】出力形式の指示(format instruction)
連携先ツールのインターフェースに合わせるよう
LLM出力形式をプロンプトで指示

プロンプト①（インプットデータの生成）

故障情報（トラチケRAGの要約テキスト）
奈良FTEDGE#1にて故障がありました。
10:03 ALM発生し 15:00 対応完了しています

質問
故障した装置のホスト名は？

出力形式
以下のformatのJSON形式で出力してください
JSONのみ出力し余計なテキストを含めないでください
{ "host_name": "NW装置のホスト名" }

LLM出力①

```
{ "host_name": "奈良FTEDGE#1" }
```

LLMが出力したパラメータをBodyにセットして
HTTP Request送信



NW可視化ツール

構成計算ロジック
作図ライブラリ

plotly

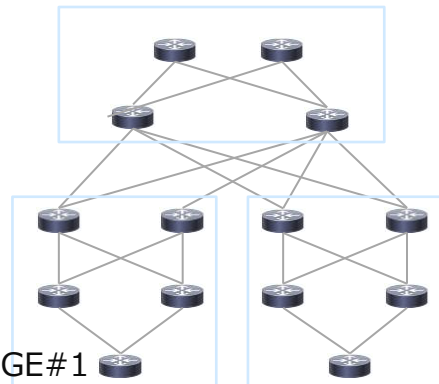


周辺装置の接続情報など
NW構成情報を参照



LLM出力②

奈良FTEDGE#1にて故障がありました。
10:03 ALM発生し 15:00 対応完了しています
下記に関連装置の構成を示します



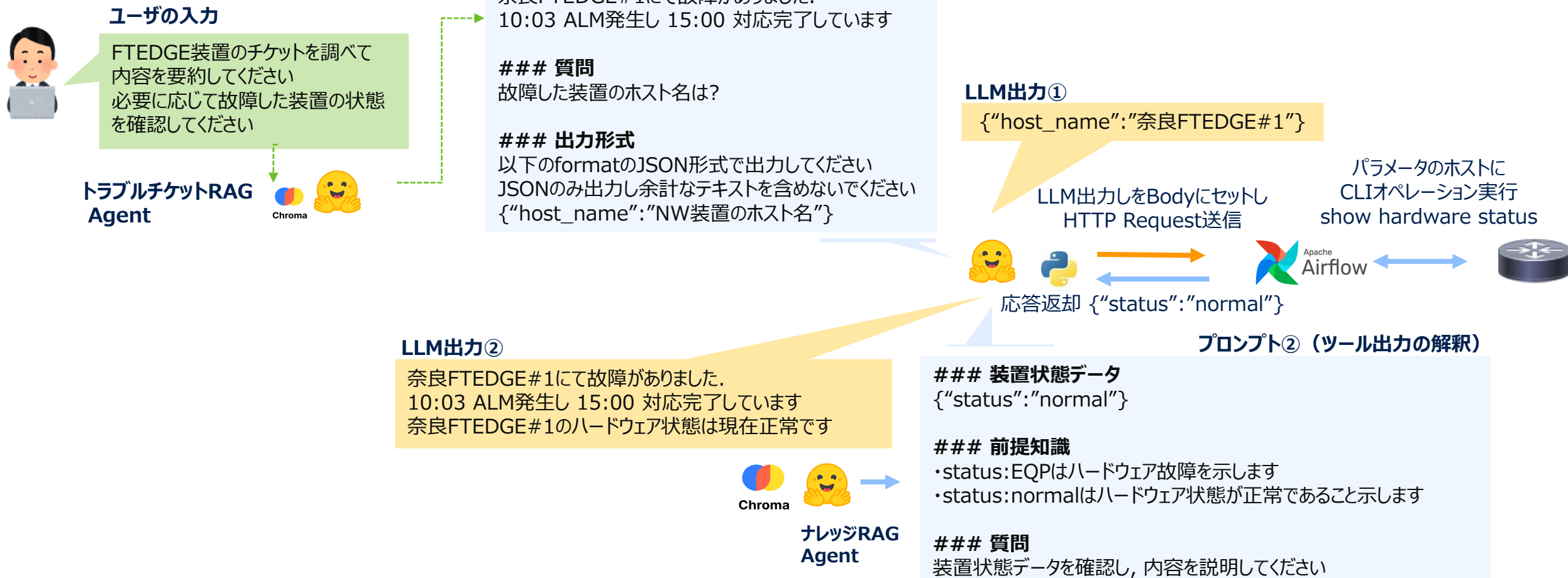
奈良FTEDGE#1

GUIに出力

オペレーションツールとLLM連携事例紹介②：ワークフローエンジンとの連携

- 先の事例と同様に必要な入力データをLLMに生成させ、また応答をLLMに解釈させることでワークフローエンジンとの連携を通してLLMからNW装置へのCLIオペレーションの実行が可能となる。
- さらに動作の正確性を向上させるために、出力の解釈のプロンプトには**前提知識の挿入**のチューニングが必要となる。

ワークフローエンジン連携Agentの構成例



ツールデモ（画面キャプチャ・動画など） 実施予定

まとめと今後の課題

- Local LLMを中心としたアーキテクチャと, RAG・Agentの工夫により, トラブルチケットに対する単純なQAや, ワークフローエンジンとの連携によるCLIオペレーション実行など, **NWオペレーションの基本的な動作にLLMを活用**することができ, 今後のユースケースの拡大に向けた技術検討を進捗させることができた.
- 実運用に資する（オペレータが便利さを体感できるレベル）ためには, **専門的な生成・推論を伴うQAや, 複数のオペレーションツールとの連携**を行うための技術方式をブラッシュアップする必要性が判明した.
- 特化したタスクを担う複数のAgentを連携させる**マルチエージェント**が有力なアプローチと考えており, アーキテクチャの具体化を進めていく予定.

- NWオペレーションにおいて, 生成AIを活用したいユースケースについて
- 生成AI活用の際する課題感について (セキュリティや環境構築のハードルなど)
- 紹介した技術スタック及びフレームワーク (RAG・langchainなど)に関する議論 など